

## **ЗАДАНИЕ 2 ДЛЯ САМОСТОЯТЕЛЬНОЙ ПРАКТИЧЕСКОЙ РАБОТЫ ПО ДИСЦИПЛИНЕ «ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА БАЗ И ХРАНИЛИЩ ДАННЫХ»**

Построить модель системы на основе методологии DFD и разработанной функциональной модели предметной области IDEF0.

Модель должна включать в себя:

- контекстную диаграмму;
- диаграммы декомпозиции (на диаграмме декомпозиции отобразите внешние сущности и хранилища).

### ***2.1. Методология DFD***

Целью методологии является построение модели рассматриваемой системы в виде диаграммы потоков данных (Data Flow Diagram – DFD). Диаграммы потоков данных предназначены прежде всего для описания документооборота и обработки информации, хотя допускают и представление других объектов.

При создании диаграммы потоков данных используются четыре основных понятия:

- потоки данных,
- процессы (работы) преобразования входных потоков данных в выходные,
- внешние сущности,
- накопители данных (хранилища).

*Потоки данных* являются абстракциями, используемыми для моделирования передачи информации (или физических компонент) из одной части системы в другую. Потоки на диаграммах изображаются именованными стрелками, ориентация которых указывает направление движения информации.

*Процессы (работы)* служат для преобразования входных потоков данных в выходные. Имя процесса должно содержать глагол в неопределенной форме с последующим дополнением (например, «получить документы по отгрузке продукции»). Каждый процесс имеет уникальный номер для ссылок на него внутри диаграммы, который может использоваться совместно с номером диаграммы для получения уникального индекса процесса во всей модели.

*Хранилище (накопитель)* данных моделирует данные, которые будут сохраняться в памяти между процессами. Информация, которую содержит хранилище, может использоваться в любое время после ее получения, при этом данные могут выбираться в любом порядке. Имя хранилища должно определять его содержимое и быть существительным.

*Внешняя сущность* представляет собой материальный объект вне контекста системы, являющейся источником или приемником данных. Ее имя должно содержать

существительное, например, «склад товаров». Предполагается, что объекты, представленные как внешние сущности, не должны участвовать ни в какой обработке.

Кроме основных элементов, в состав DFD входят словари данных и мини спецификации.

Словари данных являются каталогами всех элементов данных, присутствующих в DFD, включая потоки данных, хранилища и процессы, а также все их атрибуты. Мини спецификации обработки – описывают DFD-процессы нижнего уровня. Фактически мини спецификации представляют собой алгоритмы описания задач, выполняемых процессами: множество всех мини спецификаций, является полной спецификацией системы.

Диаграммы потоков данных строятся в виде иерархии. Сначала строится контекстная диаграмма. При проектировании относительно простых систем строится единственная контекстная диаграмма со звездообразной топологией, в центре которой находится так называемый главный процесс, соединенный с приемниками и источниками информации, посредством которых с системой взаимодействуют пользователи и другие внешние системы. Перед построением контекстной DFD необходимо проанализировать внешние события (внешние сущности), оказывающие влияние на функционирование системы. Количество потоков на контекстной диаграмме должно быть по возможности небольшим, поскольку каждый из них может быть в дальнейшем разбит на несколько потоков на следующих уровнях диаграммы.

Для сложных систем (признаками сложности могут быть наличие большого количества внешних сущностей (десять и более), распределенная природа системы или ее многофункциональность) строится иерархия контекстных диаграмм. При этом контекстная диаграмма верхнего уровня содержит не единственный главный процесс, а набор подсистем, соединенных потоками данных. Контекстные диаграммы следующего уровня детализируют контекст и структуру подсистем.

Для проверки контекстной диаграммы можно составить список событий. Список событий должен состоять из описаний действий внешних сущностей (событий) и соответствующих реакций системы на события. Каждое событие должно соответствовать одному или более потокам данных: входные потоки интерпретируются как воздействия, а выходные потоки – как реакции системы на входные потоки.

Каждый процесс на DFD, в свою очередь, может быть детализирован при помощи DFD или (если процесс элементарный) спецификации. Спецификация процесса должна формулировать его основные функции таким образом, чтобы в дальнейшем специалист, выполняющий реализацию проекта, смог выполнить их или разработать соответствующую программу. Спецификация является конечной вершиной иерархии DFD. Решение о завершении детализации процесса и




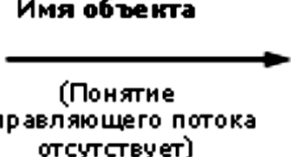

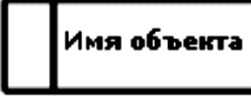

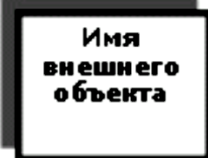
использовании спецификации принимается аналитиком исходя из следующих критериев:

- наличия у процесса относительно небольшого количества входных и выходных потоков данных (2-3 потока);
- возможности описания преобразования данных процессов в виде последовательного алгоритма;
- выполнения процессом единственной логической функции преобразования входной информации в выходную;
- возможности описания логики процесса при помощи спецификации небольшого объема (не более 20-30 строк).

В качестве языка спецификаций обычно используются структурированный естественный язык или псевдокод.

В методологии DFD используются две нотации: Йодана-Де Марко (Yourdan) и Гейна-Сарсона (Gane-Sarson) (табл. 1.1).

Таблица 1.1. Элементы методологии DFD в нотациях Гейна-Сарсона и Йодана-Де Марко

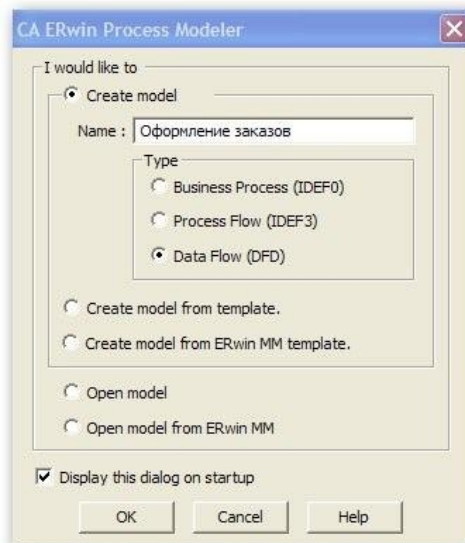
Элемент	Описание	Нотация Йордана-Де Марко	Нотация Гейна-Сарсона
<b>Функция</b>	Работа.		
<b>Поток данных</b>	Объект, над которым выполняется работа. Может быть логическим или управляющим. (Управляющие потоки обозначаются пунктирной линией со стрелкой).		
<b>Хранилище данных</b>	Структура для хранения информационных объектов.		
<b>Внешняя сущность</b>	Внешний по отношению к системе объект, обменивающийся с ней потоками.		

Следует отметить, что в ВРwin формально используется нотация Гейна-Сарсона, но с рядом отступлений: отсутствуют мини спецификации, отличается изображение

функций, контекстная диаграмма не может содержать подсистемы. Сравнивая методологии DFD и IDEF0, можно отметить, что в методологии DFD, кроме расширения изобразительных возможностей диаграмм (за счет хранилищ данных), изменяются правила интерфейсов для стрелок: стрелки могут входить и выходить с любой стороны блока.

### 2.1.1. Создание контекстной диаграммы

Методология DFD может быть использована для создания новой модели и для декомпозиции работы. Создадим новую модель работы "Оформление заказов". Для этого в диалоге создания модели выберем тип модели DFD.



Диалог создания модели

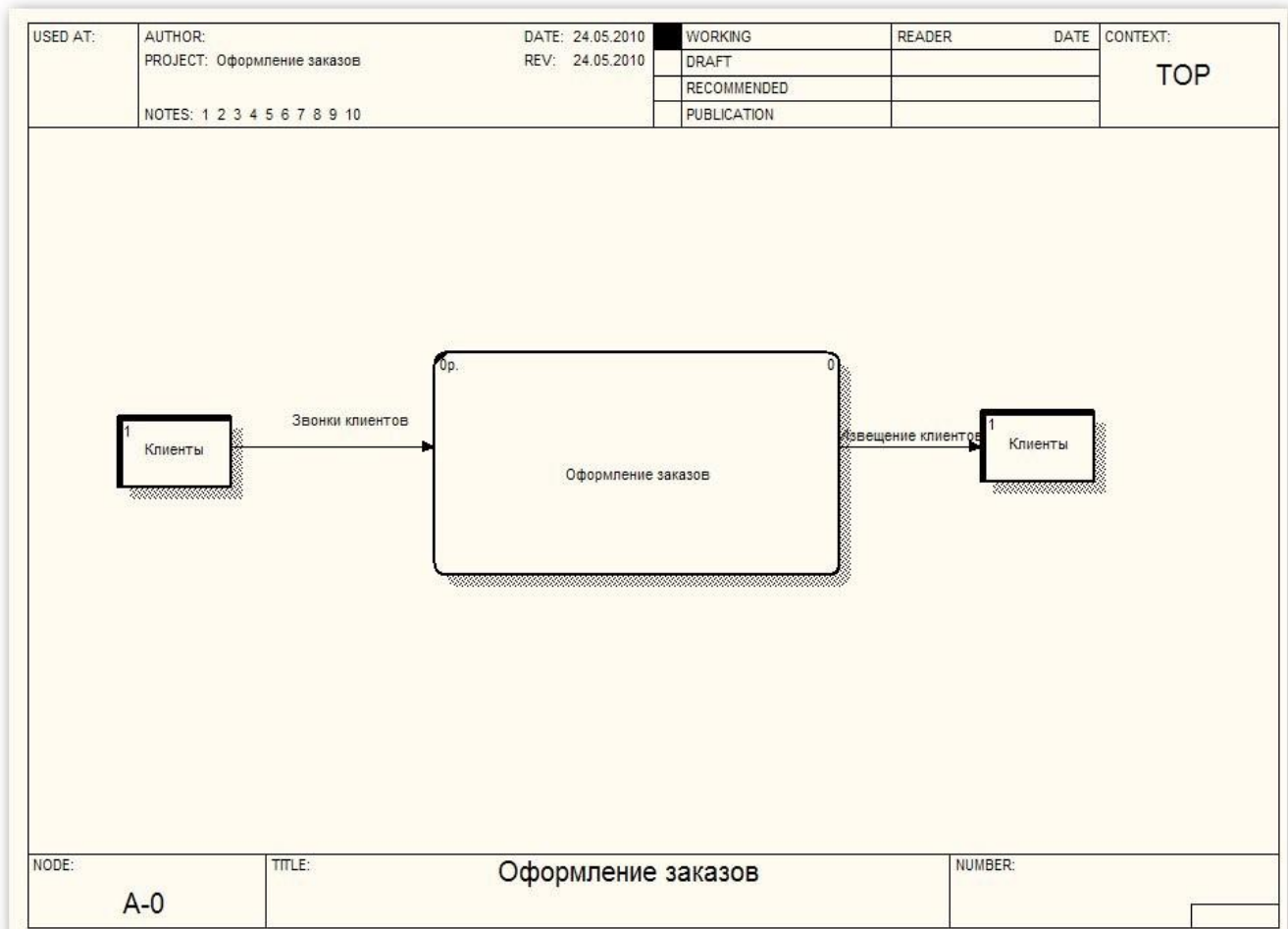
В открывшемся окне появляется единственная контекстная активность. Обратите внимание, что изображение активности немного отличается от ее изображения в методологии IDEF0: у активности закруглены углы. Зададим имя и свойства активности.

Внесем две внешние сущности: источник и приемник. В нашем случае источником и приемником будет одна внешняя сущность "Клиенты". С целью повышения наглядности покажем на диаграмме две внешние сущности с одинаковыми именами. Обратите внимание, что внешние сущности не участвуют в рассматриваемой работе и не подвергаются декомпозиции. Создается внешняя сущность с помощью

кнопки  .



Изображение внешней сущности

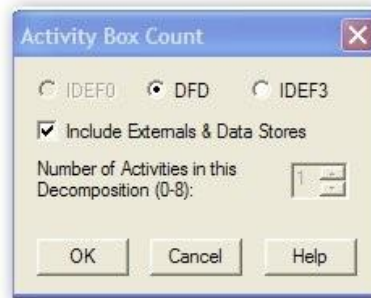


Контекстная диаграмма в методологии DFD

### ***2.1.2. Создание диаграммы декомпозиции***

Произведем декомпозицию контекстной диаграммы. Оформление заказа начинается с телефонного звонка клиента. При оформлении заказа необходимо проверить, существует ли клиент в базе данных. Если клиента нет в базе, то необходимо занести данные о клиенте в базу клиентов. Далее производится оформление и внесение заказа в список заказов. При оформлении заказа используются как база клиентов, так и список продуктов. Заканчивается оформление заказа извещением по телефону клиента о результатах оформления заказа (можно было бы включить и уточнение заказа). Таким образом, в простейшем случае декомпозиция будет включать две активности: "Проверка и внесение клиента" и "Внесение заказа".

При создании диаграммы декомпозиции в диалоге Activity Box Count следует выбрать тип диаграммы декомпозиции (IDEF0 выбрать нельзя).



Диалог Activity Box Count для методологии DFD

Свойство Include Externals & Data Stores означает, что на дочернюю диаграмму будут мигрировать внешние сущности и хранилища данных с родительской диаграммы. При этом родительская диаграмма копируется на дочернюю. На дочерней диаграмме надо удалить родительскую активность и создать необходимое число активностей. При удалении активности на дочерней диаграмме необходимо разрешить туннелированные стрелки на родительской диаграмме. Если это свойство не включено, то внешние сущности и хранилища не мигрируют на дочернюю диаграмму; можно также задать число сущностей на дочерней диаграмме. В нашем примере выберем миграцию внешних сущностей и хранилищ (на нашей родительской диаграмме нет хранилищ, но в принципе они возможны).

Построим диаграмму декомпозиции.

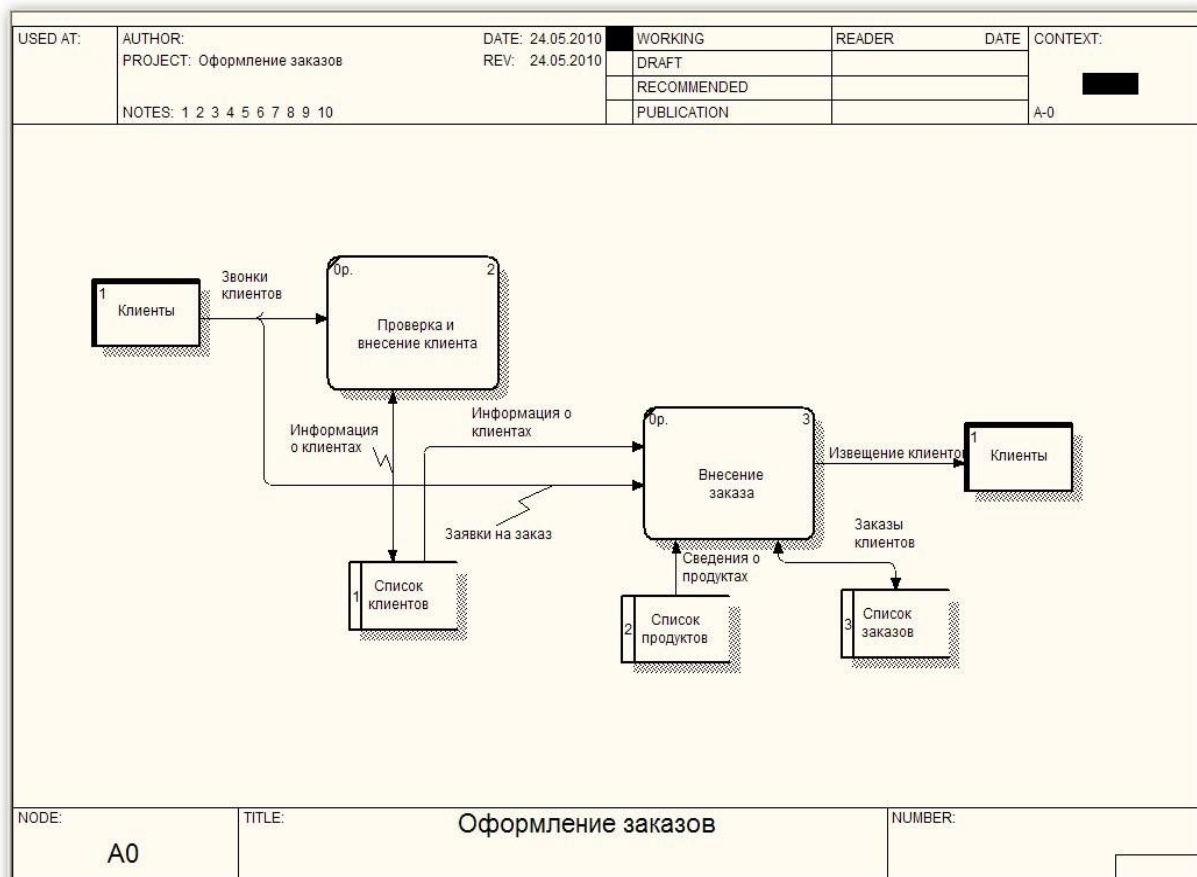



Диаграмма декомпозиции в методологии DFD

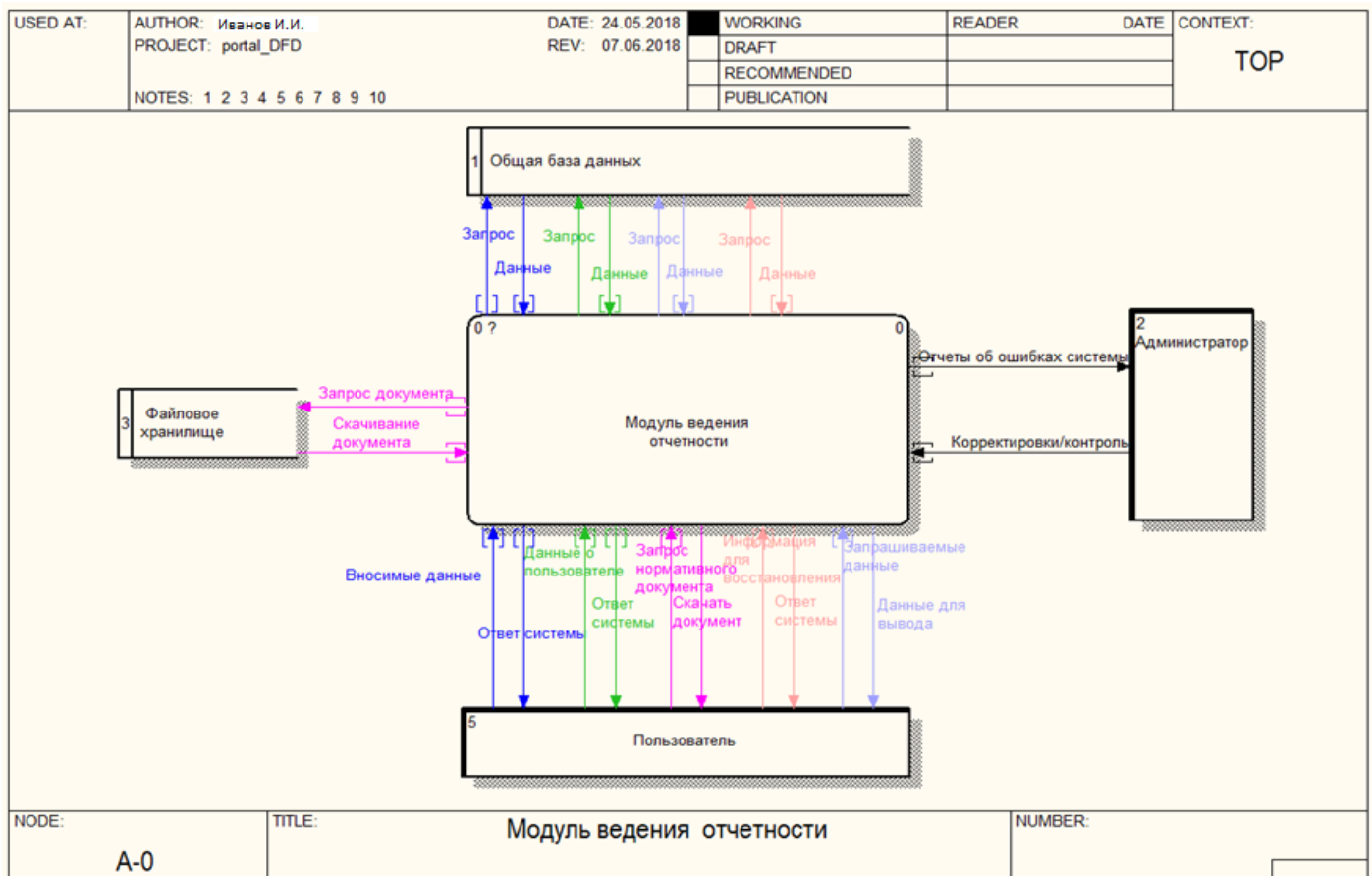
На диаграмме декомпозиции присутствуют хранилища, которые создаются с помощью кнопки .



Изображение хранилища

Некоторые стрелки на диаграмме декомпозиции являются двунаправленными. Сначала изображается однонаправленная стрелка. Чтобы сделать стрелку двунаправленной, щелкните правой кнопкой мыши по стрелке, из контекстного меню выберите пункт Style и на вкладке Style меню свойств стрелки выберите вариант двунаправленной стрелки (Bidirectional).

### 2.1.3. Пример моделирования системы на основе методологии DFD



Контекстная диаграмма в методологии DFD модуля ведения отчетности



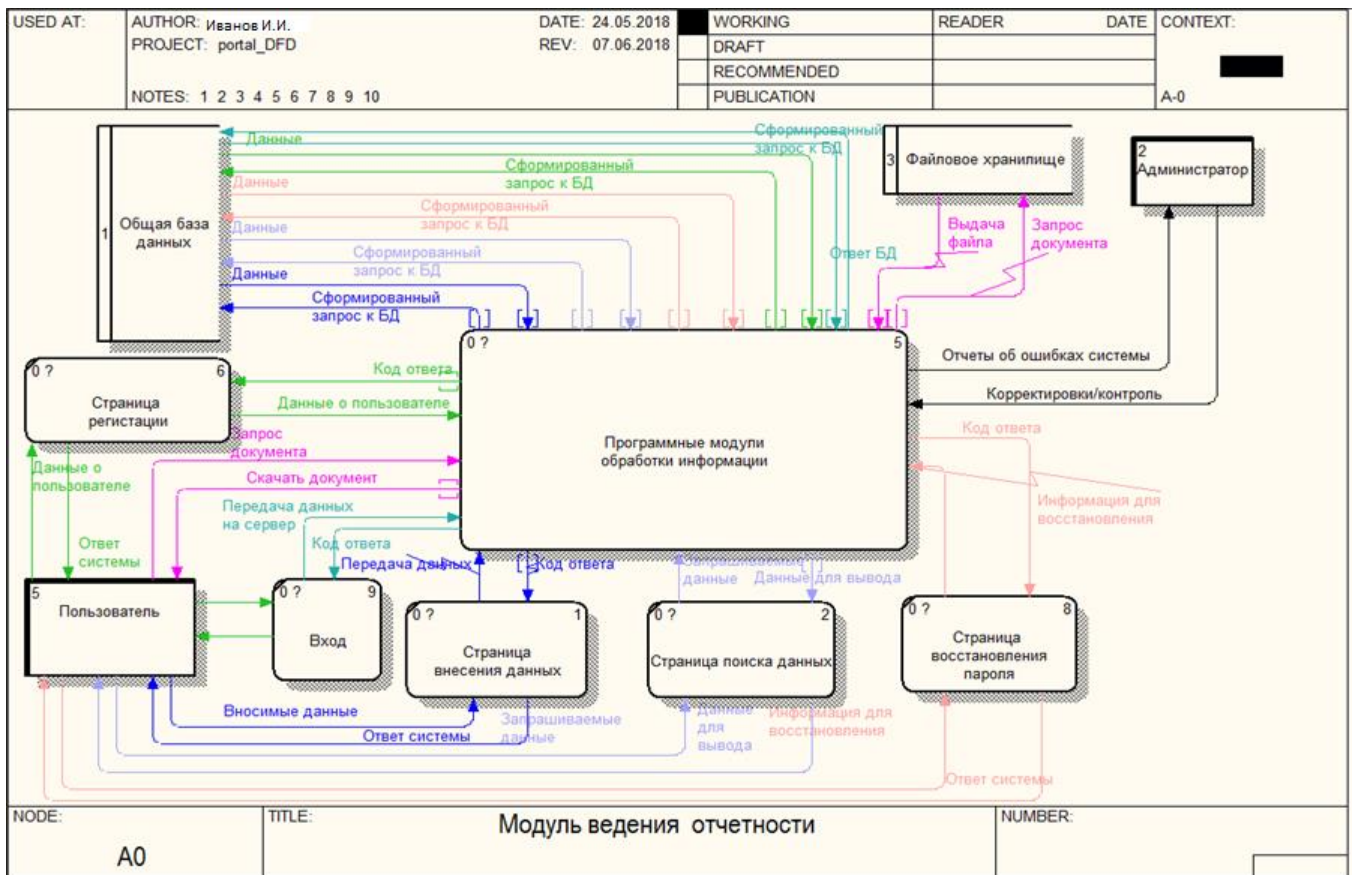


Диаграмма декомпозиции в методологии DFD модуля ведения отчетности

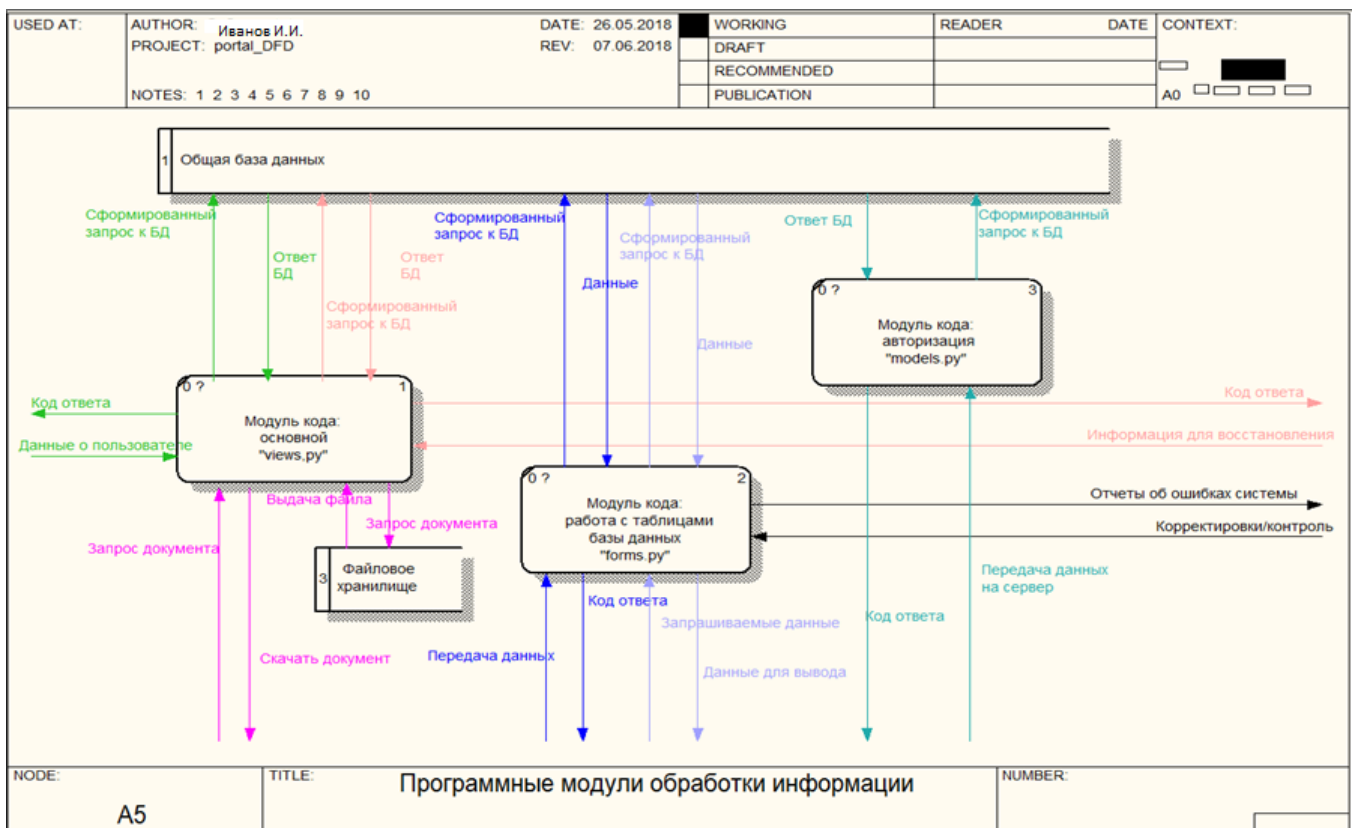


Диаграмма декомпозиции в методологии DFD процесса Программные модули обработки информации